

## 北京工业大学 2001 年数据结构试题及答案

### 一. 多选/填空题。

- 一个栈的入站元素序列是 1, 2, 3, 4, 5 若允许出栈操作可在任意可能的时刻进行, 则下面的序列中。不可能出现的出栈序列是 ( ), 理由是 ( )。  
A. 3, 4, 2, 5, 1  
B. 2, 5, 4, 1, 3  
C. 2, 3, 1, 5, 4  
D. 3, 5, 4, 2, 1
- 一棵二叉树的前序遍历序列为 ABCDEFG, 它的中序序列可能是 ( )  
A. CABDEFG  
B. ABCDEFG  
C. DACEFBG  
D. BADCFEG
- 下面结构中最适于表示稀疏无向图的是 ( ), 适于表示有向图的是 ( )  
A. 邻接矩阵  
B. 逆邻接表  
C. 邻接多重表  
D. 十字链表  
E. 邻接表
- 采用败者树进行 K 路平衡归并时, 总的 (包括访外) 效率与 K ( )  
A. 有关  
B. 无关
- N 个顶点连通无向图的邻接矩阵至少有 ( ) 个非 0 元素, 至多有 ( ) 个非 0 元素;  
n 个顶点的强连通有向图至少有 ( ) 条弧。
- 含 4 个度为 2 的结点和 5 个叶子结点的二叉树, 可有 ( ) 个度为 1 的结点。

### 二. 简答题

- 上三角阵 A (N\*N) 按行主序压缩存放在数组 B 中, 其中  $A[I,j]=B[K]$ . 写出用 I, J 表示的 K。
- 画出广义表  $A=(a, (b, ()), (((), c))$  的第一种存储结构 (表结点第二指针指向余表) 图, 并用取首元 (head()) 和取尾元 (tail()) 函数表示原子 c。
- 证明: 在二叉树的三种遍历序列中, 所有叶子结点间的先后关系都是相同的。要求每步论断都指出根据。
- 是述哈希表中不成功的平均查找长度概念, 求法, 和理由。
- 对于输入关键字序列 48, 70, 65, 33, 24, 56, 12, 92 进行:
  - 建立堆排序的初始堆 (小顶堆), 要求画出主要过程。
  - 建一棵平衡二叉树, 画出过程 (至少每次调整有一张, 标出最小不平衡子树的根)。

### 三. 下面程序段实现用尾指针表示, 带头结点单链环的操作, 请补足空缺部分。

```
typedef struct node{
    Elemtype key;
    Struct node *next;
    Node,*link;
```

```
program ppp;
type elemtype=char;
link=^node;
node=record
    Key:elemtype; next:link;
```

```
Oid init(llink &p)      }
```

```
end;
```

## C 语言

```
Void ins(link &p,int I,elemtype e) {  
    int j; link q,s;q=p->next; j=0;  
    while (q!=p && j<I-1){q=q->next; j++;}  
    if (②填空) exit(error)  
    s=(link)malloc(sizeof(node));  
    s->key=e; s->next=q->next=s; q->next=s; /*维护*/  
    ③填空  
    return;  
}  
void del(link &p,int I,elemtype &e) {  
    int j; link q,s;  
    q=p->next j=0;  
    while (q!=p && j<I-1){q=q->next; j++;}  
    if (④填空) exit(error)  
    s=q->next; q->next=s->next; e=s->key; /*维护*/  
    ⑤填空  
    free(s); return;  
}  
find (link p,elemtype e) {  
    int I; link q;  
    q=p->next; ⑥填空; q=q->next; I=1;  
    while (⑦填空) {q=q->next; I++;}  
    if (q=p->next) return 0;  
    else return I;  
} /*循环条件中只有一次比较*/
```

## 类 PASCAL 语言

```
proc init(var p:link);  
begin ①填空 end;  
proc ins(var p:link; I:integer; e:elemtype);  
var j:integer; q,s:link;  
begin q:=p^.next; j:=0;  
    while (q<>p) and (j<I-1) do  
        [q:=q^.next;j:=j+1]  
    if ②填空 then exit(error);  
    new(s); s^.key=e;  
    s^.next:=q^.next; q^.next:=s; /*维护*/  
    ③填空  
end;
```

```

proc del(var p:link; I:integer; var e:elemtype);
var j:integer;q,s:link;
begin q:=p^.next; j:=0;
  while (q<>p) and (j<I-1) do
    [q:=p^.next; j:=j+1];
    if ④填空 then exit(error);
    s:=q^.next; q^.next=s^.next; e:=s^.key; ⑤填空; /*维护*/
    dispose(s)
  end;
end;
func find(p:link; e:elemtype):integer:
  q:link; I:integer;
begin q:=p^.next; ⑥填空;
  q:=q^.next; I:=1;
  while ⑦填空 do
    [q:=q^.next; I:=I+1];
    if (q=p^.next) then return (0)
    else return(i)
  end; /*循环条件中只有一次比较*/

```

## 北京工业大学 2001 年研究生入学考试试题答案

一. 单选/多选和填空题, 每空 2 分, 共 20 分

1. 〈1〉 B 〈2〉 1 比 3 先进栈, 应后出栈。
2. 〈3〉 BD
3. 〈4〉 C 〈5〉 BDE
4. 〈6〉 A
5. 〈7〉  $2 * (n-1)$  〈8〉  $n * (n-1)$  〈9〉  $n$
6. 〈10〉 任意多

二. 简答题 25 分

1. (5 分)

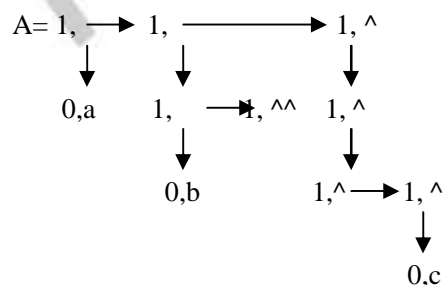
用  $I, j$  表示  $k$ :  $k = (n+n-I-2)(I-1)/2 + j - I = (2n-I)(I-1)/2 + j$ .

2. (5 分)  $c = \text{head}(\text{tail}(\text{head}(\text{head}(\text{tail}(\text{tail}(A))))))$

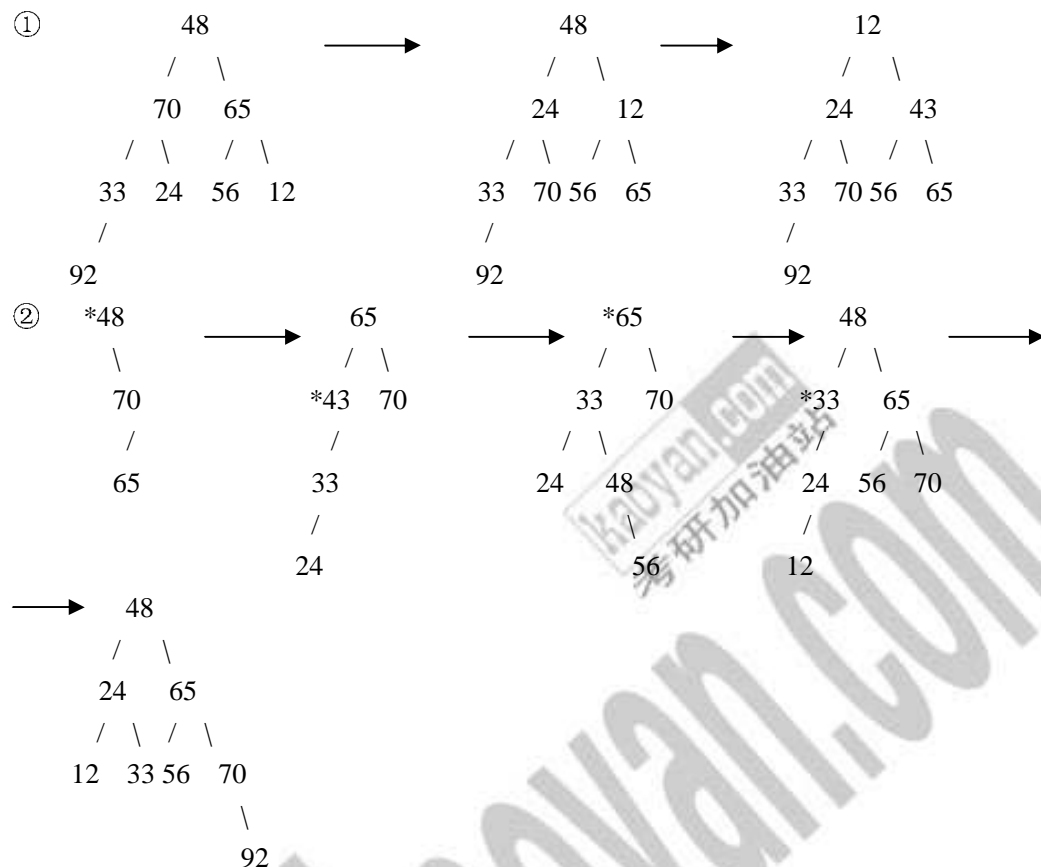
3. (5 分) 取任意两个叶结点  $u, v$ , 它们同属于一棵二叉树, 必有共同祖先, 记其中最近的为  $w$ ,  $u, v$  不会是  $w$ , 若是就不可能为叶子; 故  $u, v$  分属  $w$  的左右子树, 设  $u$  在左, 则按定义, 在三种遍历序列中,  $u$  都在  $v$  前面。由  $u, v$  的任意性可知, 所有叶子结点的先后关系都是相同的。

4. (4 分) 哈希表中不成功的平均查找长度概念和求法指: 从每个可能的哈希地址开始按算法约定的探测方法试探, 直至找到空闲单元为止, 其间进行比较的次数即为该地址的不成功查找长度。所有可能的哈希地址的不成功查找长度的平均值, 就是哈希表的不成功平均查找长度 (2 分)。不成功查找对应的关键字可能是无穷的, 但映射到每个哈希地址都是可能的, 因此, 在没有先验知识的情况下, 认为它们映射到每个哈希地址的概率相等是合理的假设 (2 分)。

二.2 图



5. (6分)



三. 程序填空, 每空 3 分, 共 21 分

## C 语言

- ① `p=(link)malloc(sizeof(node));`  
    `p->next=p;`
- ② `I<1; j<I-1`
- ③ `if(q==p) p=s;`
- ④ `I<1; q==p` 注意与②不同!!
- ⑤ `if(s==p) p=q;`
- ⑥ `q->key=e;`
- ⑦ `q->key!=e`

## 类 PASCAL 语言

- ① `new(p); p^.next:=p;`
- ② `(I<1)or(j<I-1)`
- ③ `if(q=p) then p:=s;`
- ④ `(I<1)or(q=p)`
- ⑤⑥ `if(s=p) then p:=q;`
- ⑥ `q^.key:=e;`
- ⑦ `q^.key<>e`

四. 程序将单链表逆置 (4 分), 存入一个带头结点, 用尾指针表示的单向循环链表 (2 分)。算法时间复杂度为  $O(n)$ , 每递归调用一层即进入下一结点, 调用深度与表的相当, 退出时处理插入, 每层复杂度为  $O(1)$ , 故总的复杂度为  $O(n)$ 。(2 分)

五. (10 分)

## C 语言

```
typedef char elemtype;
type def struct node{
elemtype key;
struct node *f,*b; /*f 为孩子指针*/
}node,link;
link t;
pp(link p){
    if (!p) return 0;
    if (!p->b) {
        printf("%c",p->key);
        return pp(p->f)+1;
    }
    else return pp(p->f)+pp(p->b);
}
main(){
    int j;
    建根指针 t 指出的森林的二叉树表示;
    j=pp(t);
    printf("\nNumj=%d\n",j);
}
```

## 类 PASCAL 语言

```
program ppp;
type elemtype=char;
link=^node;
node=record
    key:elemtype;
    f,b:link;
end;
var t:=link; j:=integer;
func pp(p:link):integer;
begin If p=nil then return(0)
    if p^.b=nil then write (p^.key);
    return (pp(p^.f)+1)
    else return (pp(p^.f)+pp(p^.b))
end;
```

begin

建根指针 t 指出的森林的二叉树表示;

j:=pp(T);

writeln;writeln('Num=',j)

end.

评分标准: 结构定义 2 分, 统计兄弟指针为空 2 分, 递归算法 (公共变量-过程亦可) 4 分, 正确输出 2 分

六. (16 分) 根据右表给出的顶点数, 边数, 顶点信息, 弧的信息  
(边, 权) 按在链头插入的算法:

1. (6 分) 画出 AOE 网的邻接表结构图, 并用类 C (或类 PASCAL) 描述类型。

2. (4 分) 按结构图和求关键路径的算法写出顶点的拓扑排序序列, 估算拓扑排序算法的时间复杂度。

3. (6 分) 求出各弧代表的活动的最早开始时间和最迟开始时间, 指出关键活动。

1. 0 A → 4, 4 → 3, 2 → 1, 3 ^

1 B → 2, 1 ^

2 C → 6, 3 → 5, 1 ^

3 D → 2, 3 ^

4 E → 7, 2 ^

5 F → 8, 5 ^

6 G → 8, 4 ^ 邻接表结构图

7 H → 6, 2 ^

8 I ^

2. ABDCEHGI o(e+n)

3. e ee el mark

AB 0 1

AD 0 0 \*

AE 0 0 \*

BC 3 4

DC 2 2 \*

EH 4 4 \*

CF 5 6

CG 5 5 \*

HG 6 6 \*

FI 6 7

GI 8 8 \*

六. 1 类型定义

## C 语言

```
#define maxvnum 20
```

```
typedef struct arctype {
```

```
int headnum, cost;
```

```

struct arctype *next;
}arctype,*link;
typedef struct {
elemtype key;
link firsttout;
}vextype;
typedef struct {
vextype vex[maxvnum];
int vexnum,arcnum;
}adjlist;

```

## 类 PASCAL 语言

```

const    maxvnum=20;
type
  link=^arctype;
  arctype=record
      headnum,cost: integer;
      next: link
  end;
  vextype=record
      key: elemtype;
      firsttout: link
  end;
  adjlist=record
      vex: array[1..maxvnum] of vextype;
      vexnum,arcnum: integer
  end;

```

四. (8 分) 阅读下面的程序, 指出过程 pp 完成的功能及结果数据结构的名称, 并估计算法的时间复杂度  $O(?)$ , 说明理由. 设单链表长度为 n.

## C 语言

```

typedef char elemtype;
typedef struct node{
    Elemtype key;
    Strcut node *next;
}node,*link;
link la ;
void pp(link p) {
    if (p) {
        pp(p->next); p->next=la->next;
        la->next=p;la=p;
    }
}

```

```

    } return;
}
main () {
link q;
    建立用 la 指出的带头结点的单链表;
    q=la->next; la->next=la; pp(q);
    输出用 la 指出的链式结构的数据元素;
    return 1;
}

```

## 类 PASCAL 语言

```

Program ppp;
Tyoe elemtype=char;
    Link=^node;
    Node=record
        Key:elemtype;
        Next:link;
    End;
Var la,q:link;
Proc  pp(p:link);
Begin
    If (p<>nil) then
        Begin  pp(p^.next);
            P^.next:=la^.next;
            La^.next:=p; la:=p
        End
    End;
Begin
    建立用 la 指出的带头结点的单链表;
    q:=la^.next; la^.next:=la; pp(q);
    输出用 la 指出的链式结构的数据元素;
end.

```

- 五. (10 分) 编程打印出用孩子兄弟链表表示的森林中最小兄弟  
(无弟弟者) 结点并统计输出其个数。设结点数据域为字符 (字母),  
要求描述所用结构。
- 六. (16 分) 根据右表给出的顶点数, 顶点信息, 弧的信息按在链头插入的算法:
- (6 分) 画出 AOE 网的邻接表结构图, 并用类 C (或用类 pascal) 描述类型。
  - (4 分) 按结构图和求关键路径的算法 (!) 写出顶点的拓扑排序序列,  
估计拓扑排序算法的复杂度。
  - (6 分) 求出各弧代表的活动的最早开始时间和最迟开始时间, 指出关键活动。

9 11

ABCDEFGHI



AB 3  
AD 2  
AE 4  
BC 1  
DC 3  
EH 2  
CF 1  
CG 3  
HG 2  
FI 5  
GI 4